

## 階層的コントロールの論理とプログラム抽出

著者	亀山 幸義
発行年	2013
その他のタイトル	Logical aspect of Control Operators and Program Extraction
URL	<a href="http://hdl.handle.net/2241/120878">http://hdl.handle.net/2241/120878</a>

# 科学研究費助成事業（学術研究助成基金助成金）研究成果報告書

平成25年5月24日現在

機関番号：12102

研究種目：挑戦的萌芽研究

研究期間：2011～2012

課題番号：23650003

研究課題名（和文） 階層的コントロールの論理とプログラム抽出

研究課題名（英文） Logical aspect of Control Operators and Program Extraction

研究代表者

亀山 幸義 (KAMEYAMA YUKIYOSHI)

筑波大学・システム情報系・教授

研究者番号：10195000

研究成果の概要（和文）：プログラムの実行順序を制御するコントロールオペレータについて、必要呼び計算と並列計算の2つの設定で、論理的な側面の研究を行った。前者においては、論理型プログラミング言語のある種の探索機能と関数型プログラミング言語の限定継続コントロールオペレータが対応することを示した。後者においては、適切なコントロールのもとで、コントロールオペレータを持つ並列計算体系が透過的な意味論を持つことを証明した。

研究成果の概要（英文）：We study delimited-control operators in the context of call by need evaluation and parallel computation from logical viewpoints. We have shown that, encapsulated search in a certain kind of logic (or functional logic) programming languages is simulated by delimited-control operators in call-by-need functional programming languages. We have also proved that under a suitable restriction, parallel programming languages with control operators have transparent semantics, which means that adding or eliminating parallel primitives has no impact on the meaning of programs.

交付決定額

（金額単位：円）

	直接経費	間接経費	合 計
交付決定額	1,800,000	540,000	2,340,000

研究分野：プログラム論理

科研費の分科・細目：情報学・情報学基礎

キーワード：プログラム理論

## 1. 研究開始当初の背景

型付きラムダ計算と直観主義論理の体系の間の密接な対応関係を表す Curry-Howard の同型対応は、1990 年頃から、コントロールオペレータを持つ計算体系と古典論理の体系の対応へと拡張され、様々な研究がなされてきた。コントロールオペレータの中で近年もっともよく研究されている限定継続コントロールオペレータは、種々の計算エフェクトを表現できるという、ある種の万能性を持っているため特に興味深く、それに対応する論理が何であるのかも研究者の関心を集めていた。そのような論理体系が構築されれば、いわゆる「プログラム抽出」の技法によ

り、仕様の証明から、限定継続コントロールオペレータを持つプログラムが自動的に作成できることになる。

限定継続コントロールオペレータは、一定の範囲のすべての計算エフェクトを表現することができるが、それを他の計算エフェクトと階層的に組み合わせたときの振る舞いは十分には知られていない。計算エフェクトをモナドで表現し、計算エフェクトの組み合わせを階層的なモナドで表現する方法はあるが、階層的モナドだけですべての組み合わせを表現できるとは限らない。これらについて論理学の見地から適切な位置付けをあたえることは、一般論としては困難ではあるが、

いくつかの有用なケースで達成できれば、コントロールオペレータに対する重要な貢献となることが予想された。

## 2. 研究の目的

本研究の目的は、限定継続コントロールオペレータやそれ以外の計算エフェクトを起こす機能を階層的に組み合わせた計算体系について、対応する論理体系が何であるかを解明すること、また、それによって限定継続コントロールオペレータについて新しい知見を得て、さらにはプログラム抽出につなげることである。

階層的な組み合わせとしてまず考えられるのは、限定継続のみの階層化であるが、これは、本研究代表者による過去の研究で、等式系による意味論や CPS 変換などが得られており、対応する論理体系についても「準古典論理」という形で得られていた。そこで、本研究では、限定継続と他の計算エフェクトの階層的な組み合わせについて検討することを目的とした。他の計算エフェクトとして、現実的な計算環境で有用性の高い並列化プリミティブと、基礎研究が近年活発化している必要呼び計算という、両極端の2つを選択し、階層化コントロールのさまざまな側面を観察することも目的とした。

## 3. 研究の方法

(1) 必要呼び計算体系と限定継続コントロールオペレータの階層化について、検討を行った。いくつかある必要呼び計算体系の標準的な定式化について、限定継続コントロールオペレータを追加した体系が、適切な計算体系として必要な性質を満たすか検討した。また、その体系が名前呼び計算や値呼び計算とどう関係するかを調べた。

次にこれらの体系に対する型システムについて考察し、それが形作る型の世界に対応する論理体系がどのようなものであるか考察した。

(2) 本研究課題の補助金により筑波大学を訪問していた Shan 研究員、および、Kiel 大学のグループと連携して研究を進めた。上記の体系に対応する論理体系そのものより、上記の体系（関数型プログラミング言語の計算体系）と、論理型プログラミング言語の対応が非常に興味深いことが判明し、目標をそちらに切り替えた。すなわち、論理型（正確には、関数論理型）プログラミング言語における「カプセル化した探索」を表すプリミティブと、関数型プログラミング言語における限定継続コントロールオペレータを対応付けるものである。これにより、前者の不明確な意味論が明確化され、たとえば Curry といったプログラミング言語の意味論への重要な

貢献となる。

(3) 並列計算における限定継続コントロールオペレータについては、そのような計算体系が明確でかつ論理的推論に耐える意味論を持つための必要条件である計算結果の一意性の保証について考察した。計算の並列化は本来プログラムの意味を保存すべきであるが、限定継続コントロールオペレータをもつ体系の場合、制限なく並列化すると、プログラムの意味が変わってしまう。これを避けるための計算体系を構築し、厳密に意味の保存を証明することを行った。また、そのようなプログラミング言語の実装を行った。

## 4. 研究成果

(1) 必要呼び体系として Ariola らが提案した体系を採用し、これに shift/reset と呼ばれる限定継続コントロールオペレータを追加した体系を設計した。素朴に2つの体系の和として体系を設計すると、計算の進行に伴って自由変数が発生するという問題があることを指摘し、その問題を回避する計算規則を定義した。

この体系は、計算結果の一意性、また、Ariola らの体系の保守的拡張となること、さらに、関数の引数を値に制限した場合には、値呼びの shift/reset の体系に対して保守的拡張となることを証明した。

(2) 上記の体系に対して抽象機械を定義し、名前呼び計算や値呼び計算との関係を確認した。Small step 意味論と big step 意味論のそれぞれに対応した抽象機械を設計し、それらの意味論が一致することを確認した。

また、この抽象機械に基づき、プロトタイプの処理系実装を与え、必要呼び計算と限定継続の2つが真に融合した新しい体系の提案に成功した。この処理系を使って、いくつかの比較的小さなプログラム例を走らせ、この範囲では体系における計算が適切に処理されることを確認した。

(3) 関数論理型プログラミング言語 Curry のカプセル化した探索機能に対して、ある種の限定継続コントロールオペレータをもつ必要呼び計算体系でシミュレートできることを示した。前者は、従来意味論が明確になっておらず、本研究の成果を通じて、意味論の明確化に貢献することができた。この研究の過程で、上記(1)で作成した体系は、大きなプログラム例題では必ずしも想定していたような振る舞いをしないことが判明し、例外機構と組み合わせた形の新しいタイプの限定継続コントロールオペレータを提案するに至った。シミュレーションはそちらを使って行い、適切にシミュレートできること

を確認した。このような形でコントロールオペレータが新しく定義されることは珍しく、異なる世界の橋渡しをしたことによる成果と考えられる。

(4) 並列化のためのプリミティブとして Scheme などが持つ future を採用し、future と shift/reset を持つ体系を設計した。ここでの研究のポイントは、プログラムの任意の場所を future により並列化しても、プログラムの意味は保存される、という透過性であり、この性質が常に成立するよう、shift/reset の計算規則を適切に設計することに成功した。本研究の体系は、限定継続のカプセル化の機能を使うことにより、必要以上に並列計算を阻害することがないため、call/cc と future を持つ先行研究の体系よりはるかに高い並列度が得られることが多く、有益である。

(5) この体系の透過性の証明を厳密に与えた。透過性は当然成立すると予想される性質であるが、その直感に反して、先行研究の証明をそのまま shift/reset の体系に適用するのは本質的に困難であることを発見した。これに対する解決方法として、体系の計算規則を(弱くするのではなく、逆に)強くすることにより、計算の合流性を得られるという予想外の着想に思い至り、先行研究の証明より、むしろ見通しの良い、自然な形で証明することに成功した。

透過性が厳密に保証されたことにより、本研究の体系では、プログラム中の任意の地点に future プリミティブを挿入して並列化して良いことになり、並列化の可能性が非常に大きくなった。また、証明が比較的モジュラーになり、計算体系のプリミティブごとに分離されているため、将来新しい言語プリミティブを追加する場合でも、比較的容易に、透過性の証明を拡張できるようになった。これらは、shift/reset のみならず、一般に、コントロールオペレータと future プリミティブが共存する体系にも拡張できるものであり、この分野への貢献は大きいと考えている。

(6) この項目は、本研究の当初目的には含まれていないが、研究の進展によって、新たに設定し達成した項目である。上記(5)で作成した体系が大変興味深かったため、これを実装して、プログラミング例題を構築し、プログラミング言語としての有用性を評価することとした。

本研究の体系は Scheme ベースの言語であり、CPS 変換等を通じて、並列プリミティブのみがある中間言語のプログラムに変換し、最終的に、並列計算のできる関数型プログラミング言語である Erlang のプログラムに変

換して実行する、というコンパイル方式を採用した。Erlang を採用したのは、真の並列度を測定するためと、関数型言語の特徴を利用するためである。

この枠組みで、コンパイラを作成し、本研究の体系の任意のプログラムが、Erlang プログラムとして実行できるような環境を構築した。この環境では、future は可能な限り並列に実行され、shift/reset は、通常通り、限定継続を捕捉するコントロールオペレータとして動作する。

いくつかの簡単なベンチマークプログラムをこの環境で走らせることによって、確かに並列効果があること、shift/reset と混在しても、意味論の透過性は確かに成立していること、等を確認した。

関連研究として、PLT Scheme でも future + shift/reset のプログラムが実行できる。本研究では future のバウンダリを超えて shift が限定継続を捕捉することができるが、PLT Scheme ではそのような場合は禁止されており、表現力と使いやすさの点では、本研究の体系の方が優れているといえる。一方、処理系のチューニングは行っていないので、現状では処理速度の面では十分とは言えない。この点は今後の課題となっている。

(7) 限定継続コントロールオペレータの基礎に関する種々の研究をおこなった。その成果の1つとして、限定継続とステージ化機能を持ち、.NET フレームワークとの連携機能を持つ新しいプログラミング言語を設計した。この言語では、捕捉した継続を .NET フレームワーク側の機能で可視化することができ、限定継続コントロールオペレータの研究や教育に活用することができる。

(8) 異なる限定継続コントロールオペレータの間の表現力の比較をおこなった。具体的には、再帰的データ型を持つプログラミング言語では、control/prompt というオペレータが、階層的 shift/reset や階層的 control/prompt をシミュレートできることを厳密に示した。この証明は、プログラムの意味を抽象機械により与え、抽象機械の間のシミュレーションを示したものである。限定継続コントロールオペレータのみの階層化は、表現力を本質的に高めるものではないことは従来知られていたが、それを厳密に証明することに成功した。

## 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

〔雑誌論文〕(計2件)

- ① 田中麻峰, 亀山幸義, 「限定継続機構と future を持つ計算体系の透過的意味論」, 情報処理学会論文誌, 採録決定, 2013, 査読有.
- ② Jefferson O. Andrade, Yukiyoshi Kameyama, “Efficient Multi-Valued Bounded Model Checking for LTL over Quasi-Boolean Algebras”, Special Section on Formal Approach, IEICE Transactions on Information and Systems, vol. E95-D, No. 5, pp. 1335-1364, 2012, 査読有.

〔学会発表〕(計 2 件)

- ① Sebastian Fischer, Michael Hanus, Yukiyoshi Kameyama, Chung-chieh Shan, Naoki Takashima, “Lazy Delimited Nondeterminism”, International Workshop on Functional and (Constraint) Logic Programming (WFLP 2012), Work-in-Progress session, 名古屋大学, 2012 年 5 月 29 日.
- ② 西山達也, 亀山幸義, 『コントロールオペレータを持つ必要呼び計算体系の設計』, 情報処理学会第 86 回プログラミング研究会 (PR086), 神奈川近代文学館 (横浜市), 2011 年 11 月 2 日.

## 6. 研究組織

### (1) 研究代表者

亀山 幸義 (KAMEYAMA YUKIYOSHI)  
 筑波大学・システム情報系・教授  
 研究者番号: 10195000